

A Feature Selection Method Based on Multi-objective Optimisation with Gravitational Search Algorithm

Bolou Bolou Dickson, Shengsheng Wang^(✉), Ruyi Dong,
and Changji Wen

College of Computer Science and Technology, Jilin University, Changchun
130012, People's Republic of China
boloubh@yahoo.com, wss@jlu.edu.cn

Abstract. The process of feature selection (FS) is a substantial task that has a significant effect in the performance of a given algorithm. The goal is to choose a subset of available features by eliminating the unnecessary features. This hybrid algorithm is in maximising the classification performance and minimising the number of features to achieve an outstanding performance through a less complex procedure. From the experiments, FSMOGSA was noted to be quite unparalleled in comparison with other methods in reducing the error rate, and maximising the general performance through irrelevant feature reduction.

Keywords: Gravitational search algorithm · Indexed non-dominated solutions · Pareto front · Multi-objective optimisation · Feature selection

1 Introduction

Today, the relevance of feature selection (FS) in machine learning cannot be under-rated. FS has considerable importance in real life applications such as medicine, astronomy, biology, to mention but a few. The goal is to choose a subset of available features by eliminating unnecessary features [1]. To obtain any desired result in using datasets, high dimensionality imposes learning difficulties by degradation of relevant information on the learned models. Real world datasets are entangled with many irrelevant and misleading features for this reason (FS) is adopted to eliminate such impediments. Furthermore, the objective of FS is to select a relevant subset of features say q , from a set of p features ($q < p$) in a given dataset. To extract sufficient information for example from an image set, it is appropriate to eliminate the features with no predictive information and avoid redundant features.

2 Related Works on Feature Selection

Efficient processing and retrieval of features rely on the number of relevant features extracted [2]. Hamdani et al. developed an entirely different method called, multi-objective feature selection algorithm using non-dominated sorting-based multi-objective GA II (NSGAI), however, it was not compared with any algorithm [3].

Our work is focused on Multi-Object feature selection with gravitational search algorithm (FSMOGSA) which is completely new in terms of feature selection. Tian et al. [4] proposed a work on multi-objective optimization of short-term hydrothermal scheduling using non-dominated sorting gravitational search algorithm with chaotic mutation. A. R. Bhowmik and A. K. Chakraborty, proposed, Solution of optimal power flow using non-dominated sorting multi-objective opposition based gravitational search algorithm (NSMOOGSA) [5]. And in 2013, Bing Xue et al. proposed PSO for feature selection and classification: a multi-objective approach and investigate two PSO-based multi-objective feature selection algorithms [6].

Our FSMOGSA is in maximising the classification performance and minimising the number of features to achieve an outstanding performance through a less complex method. It finds the non-dominated (Pareto fronts) solutions and groups such solutions into subsets of indexed non-dominated solutions.

2.1 Basic Gravitational Search Algorithm

Gravitational search algorithm was introduced in 2009 by Rashedi et al. [7], where the solutions of optimisation problems are regarded as agents. All agents attract one another in the solution space due to the force of gravity, lighter agents are attracted (converge) towards the heavier agents, known as the optimal solution based on the law of motion. Given a system of N agents the position of the ith agent is:

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n), \quad \text{for } i = 1, 2, \dots, N \tag{1}$$

Where x_i^d is the position of the ith agent in the dth dimension and n is the dimension of the space.

$$F_{ij}^d(t) = G(t) \frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \epsilon} (x_j^d(t) - x_i^d(t)) \tag{2}$$

Where M_{aj} is called active gravitational mass, M_{pi} is passive gravitational mass, G (t) is gravitational constant at time t, ϵ is infinitesimally small value then $R_{ij}(t)$, is a Euclidean distance between masses i and j, $R_{ij}(t) = |X_i(t), X_j(t)|_2$.

Equation (3), is the force of the object i

$$F_i^d(t) = \sum_{j=1, j \neq i}^N \text{rand}_j F_{ij}^d(t), \tag{3}$$

The acceleration of the ith object is : $a_i^d(t) = \frac{F_i^d(t)}{M_{ii}}$ (4)

2.2 Velocity and Position of Particles

The successive velocity of a given object is obtained by the addition of its current velocity to its acceleration that is Eq. (5), and the current position of the object can be obtained by (6).

$$v_i^d(t + 1) = rand_i^d v_i^d(t) + a_i^d(t) \tag{5}$$

$$x_i^d(t + 1) = x_i^d(t) + v_i^d(t + 1) \tag{6}$$

where $rand_i$ is a random number between 0 and 1, present velocity $v_i^d(t)$, next possible velocity $v_i^d(t + 1)$, next possible position $x_i^d(t + 1)$, present position $x_i^d(t)$, and acceleration $a_i^d(t)$ of the i th particle at time t .

3 Multi-objective Gravitational Search Algorithm and Pareto Front

This method operates based on the concept of dominance of set of optimal solutions called Pareto front. A given multi-objective optimisation entails maximisation or minimisation of multiple conflicting objective functions. From the training sets, any of the subsets possessing fewer features is presumed to achieve higher quality function value, as such the extrapolated features with the best fitness are chosen mostly out of such subsets of features. Reducing the number of irrelevant features will have a positive effect on the performance of the entire process. The minimisation is expressed below,

$$min, F(x) = [f_1(x), f_2(x), \dots, f_M(x)] \tag{7}$$

$$g_i(x) \leq 0, \quad i = 1, 2, \dots, m \tag{8}$$

$$h_i(x) = 0, \quad i = 1, 2, \dots, l \tag{9}$$

where x is the vector of decision variables, $f_i(x)$ is a function of x , and k is the number of objective functions to be minimised, $g_i(x)$ and $h_i(x)$ are the constraint functions of the problem. Given any minimisation task, solution x_1 will dominate solution x_2 if both satisfy this condition:

$$\forall m \in [1, M], f_m(x_1) \leq f_m(x_2) \quad \text{and} \quad \exists n : f_n(x_1) < f_n(x_2) \tag{10}$$

For $m, n \in [1, 2, \dots, M]$.

3.1 The Main Optimisation Process of (FSMOGSA)

If a given solution is not dominated by another set of solutions, then that solution is called a Pareto-optimal solution. A collection of all the sets of Pareto-optimal solutions

yields the Pareto front, some basic principles of choosing dominant or non-dominated solutions in our algorithm is based on;

- (a) The number of individuals a given individual dominates.
- (b) The Pareto front an individual is located.
- (c) The number of individuals that dominates a given individual solution.

Multi-objective tasks result if there is the necessity to make optimal decisions between two or more conflicting objectives in a solution space. Hence, we will make the equation of fitness of particles which is effective in a single objective to an equation adoptable for multi-objective as shown below:

$$M_i(t) = \|\varepsilon\| + \sum_{k=1}^k [m_i^k(t)]^2 / \sum_{j=1}^N \sum_{k=1}^K [m_j^k(t)]^2 \tag{11}$$

$$m_i^k(t) = \frac{fit_i^k(t) - worst^k(t)}{best^k(t) - worst^k(t)}, \text{ for } k \in [1, k] \tag{12}$$

where $\|\varepsilon\|$ is an infinitesimally small error value, $m_i^k(t)$ is the normalised fitness value of the i th agent in the k th objective; $fit_i^k(t)$ is the fitness value of the i th agent in the k th objective; K is the number of objectives; $best^k(t)$ is the best fitness of all agents in the k th objective; $worst^k(t)$ is the worst fitness of all the agents in the k th objective (Fig. 1).

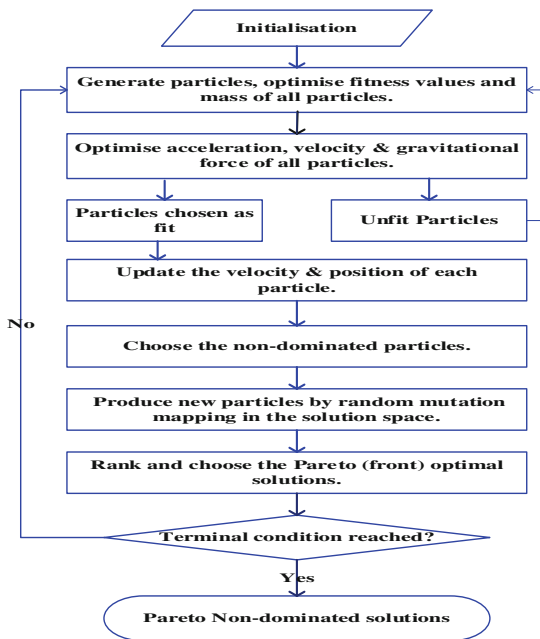


Fig. 1. Shows the FSMOGSA process.

The particles are initialised and the fitness values, velocity, acceleration and position of each particle is calculated and updated. Then, the non-dominated solutions are chosen which is followed by the random mutation to produce a new population for another optimisation.

3.1.1 The fitness function in Eq. (11) will improve the performance of the (FSMOGSA) algorithm by prudently minimising the convergence rate of agents in the process. The error factor $||\varepsilon||$ in the fitness equation stabilises the motion of the agents, it assumes infinitesimally small values within (0,1). The chosen features in the training set will be in categories of; False negative, False positive, True negative and True positive. After the fitness function had been utilised the error rate is further evaluated by Eq. (13) below:

$$ErrorRate(\psi) = \frac{Fn + Fp}{Fn + Fp + Tn + Tp} \tag{13}$$

For Fn is false negative, Fp is positive, Tn is true negative and Tp is true positive feature. This error rate can be adjusted to minimise error during feature selection.

3.1.2 Then the second purpose is reducing the number of features by choosing only very highly ranked features, redundant features are left out. This work is a multi-objective feature selection algorithm so a function other than Eq. (13) which will perform the dual purpose of the fitness function is used to minimise the classification error rate and as well guarantee minimisation of the number of features with high classification performance is adopted as in Eq. (14):

$$Fitness_{function} = \left\{ \frac{F_{selected}}{F_{All}} * \alpha + \frac{ER_{selected}}{ER_{All}} * (1 - \alpha) \right. \tag{14}$$

Where $F_{selected}$ is number of selected features, F_{All} is all the available features, α is a negligible constant within (0, 1), $ER_{selected}$ is classification error rate of the selected feature subset, ER_{All} is the classification error rate of all available features of the training set. A preponderantly negative occurrence in SI optimisation is stagnation, where the swarm agents get confined in local optimum.

3.2 Random Mutation to Generate New Agents

After every iteration period a mutation process is added to the population to randomly generate a new solution population. As a result of the unforeseen random factors, a random mapping process is employed to overcome premature convergence in FSMOGSA algorithm in the creation of new agents. Whenever a new agent dominates the current existing agent, the newly generated agent replaces the existing agent. In other words, it updates the masses and chooses the agent with heavier mass. Equations (15), (16) and (17) below are the mutation equations:

$$\zeta_i^d = [x_i^d(t) - x_{\min}^d(t)]/[x_{\max}^d(t) - x_{\min}^d(t)] \tag{15}$$

$$\eta_i^d = \lambda \zeta_i^d (1 - \zeta_i^d), \zeta_i^d \in [0, 1] \tag{16}$$

$$x \cdot c_i^d(t) = \eta_i^d [x_{\max}^d(t) - x_{\min}^d(t)] + x_{\min}^d(t) \tag{17}$$

where ζ_i^d represents the normalised position of the i th agent in the d th dimension; λ is a constant; η_i^d is the transformed value by random mutation; $x c_i^d(t)$ is the new position of the i th agent [8].

To determine the position of a particle undergoing mutation Eq. (17) is adopted. In the end of the mutation process, the steps to update the velocity and position of the offspring population is developed for ranking of the solutions then another optimisation process is carried out to select another set of Pareto solutions. The quantity ζ_i^d in Eq. (15) is randomly generated within the interval [0,1] as the process starts, λ is considered to be a constant in Eq. (16).

The mutation begins by choosing a particle say p_1 , in a random pattern within the current population. Then from a given Pareto front, another two particles p_2 and p_3 are chosen lying within a bound. By Eq. (16) the mutation factor of particle p_1 , is evaluated in the d th dimension from when d assumes the value 1 to n , then a newly mutated particle is produced. The next step is the substitution of p_1 with the newly mutated particle. When the mutation process is over the fitness values of the new population is evaluated, then, the error rate is also re-evaluated of the chosen features with Eqs. (13) and (14). Every abnormally copied code of a feature will result to a new feature (mutant), the process is not done orderly but randomly.

3.3 Indexed Non-dominated Solutions (Pareto Front Subsets)

A multi-objective method, unlike a single objective one searches for sets of optimally non-dominated solutions. The set of solutions are indexed (grouped) as non-dominated sets according to the various individual feature types. This idea helps to reduce extraneous features from the relevant features and improves the classification performance because the number of features is minimised.

Definition 3.3.1 Given different indexed feature sets, let F be a finite feature space, for every feature f in F , there is a set of features S_f such that a set of common features collected as G of S_f sets is known as an indexed collection of feature sets, that is a collection of feature sets indexed by F in different dimensions. The collection G is represented by $\{S_f\}_{f \in F}$, is the finite sets of indexed non-dominated feature sets represent Pareto fronts. Let G be the indexed sets of non-dominated solutions, so $\{S_f\}_{f \in F} = \left\{ \{S_{f_1}\}_{f_1 \in F}, \{S_{f_2}\}_{f_2 \in F}, \dots, \{S_{f_n}\}_{f_n \in F} \right\}$ hence, $\{S_f\}_{f \in F} \subset G$.

Generally, for finite sets of non-dominated solutions, the **union** and **intersection**;

$$\bigcap_{i=1}^n G_i = \{x \in U : \exists i \in \{1, 2, \dots, n\}, : x \in G_i\} \quad \text{and}$$

$$\bigcap_{i=1}^n G_i = \{x \in U : \forall i \in \{1, 2, \dots, n\}, : x \in G_i\}.$$

The idea of indexed sets of non-dominated solutions is adopted here in two aspects:

(i) Disjoint Non-dominated Solutions (Pareto Fronts) For some indexed sets of non-dominated solutions, there are some indexed finite Pareto fronts $\{G_1, G_2, \dots, G_n\}$, the arbitrary intersection of the Pareto fronts is; $G_1 \cap G_2 \dots \cap G_n = \emptyset$. It implies, $\bigcap_{f \in F} U_f = \{x | x \in U_f, \forall f \in F\} = \emptyset$, and called “*distinct or disjoint indexed non-dominated solutions*” in F which are ranked.

(ii) Connected or Intersecting Non-dominated Solutions (Pareto Fronts) If the solution subsets of non-dominated solutions have one or more common features in the indexed non-dominated solutions, then, the arbitrary intersection is non-empty. That is, $G_1 \cap G_2 \dots \cap G_n \neq \emptyset$, so $\bigcap_{f \in F} U_f = \{x | x \in U_f, \forall f \in F\} \neq \emptyset$, hence, called “*connected or intersecting indexed non-dominated solutions*”.

This segregates the non-dominated solutions and the irrelevant solutions are neglected.

The collection G is represented by $\{S_f\}_{f \in F}$, in this algorithm the finite sets of indexed non-dominated feature sets represent Pareto fronts.

3.4 K-Nearest Neighbor (K-NN) Classifier

The K-nearest neighbor (K-NN) classifier is employed here to evaluate our method as a result of its simplicity. The introduction of the K-nearest neighbor (K-NN) method in 1951 by Fix and Hodges has greatly contributed in the improvement of new algorithms. One reason of the (K-NN) algorithm is for the classification of new features after the random mutation. Due to the attributes and training samples obtained, the K-NN evaluates the classification performed by our FSMOGSA method. As a multi-objective task, the K-NN method consists of a supervised learning task where new indexed non-dominated solution sets are evaluated in the K-neighbourhood and classified based on the ranking in the solution space.

4 Experiments

The experiments were performed using some data sets from the UCI open access repository, in which three other algorithms were used to compare with the performance of our (FSMOGSA) method. Two of the algorithms are single objective method, that is, gravitational search algorithm (GSA) and Binary particles swarm optimisation (BPSO). The other is a multi-objective optimisation algorithm called Non-dominated solutions

particle swarm optimisation feature selection (NSPSOFS), these three methods were used in the comparison with our proposed method. Precisely, four different data sets were used in the experiment to ascertain the efficiency of FSMOGSA. The experiment was implemented on a 32-bit windows 7 operating system, processor: Intel®, core™, Duo 3.00 GHz, RAM: 4 GB computer, with MATLAB (R2012a) suite (Table 1).

Table 1. Is the description of the four data sets

Data sets	Iris	Ionosphere	Vehicle	Wine
No. of instants	150	351	846	178
No. of classes	3	2	4	3
No. of features	4	34	18	13

All the four methods were tested on each of the four (4) data sets and the results were compared with one another. The outcome of each of the four algorithms for each data set shows the rate of error with regards to the number of features in the data set obtained.

Table 2, shows the percentage number of features and the error values of the four methods, the best results (error values) obtained in each data set are underlined in bold character. Every data set has different number of features as such the iterations are different in number. Obviously, the error values for FSMOGSA algorithm are the least. As the number of features increases the error value increases also. The Iris data set was left out here due to the few features in it.

Table 2. Is a performance Comparison of the error values with percentage number of features.

Wine_Dataset	15%	25%	35%	45%	55%	65%	75%	85%	95%	...	%
GSA	19.312	20.114	21.089	20.400	21.822	21.997					
BPSO	17.216	18.102	23.182	24.377	24.921	23.994					
NSPSOFS	13.882	14.876	14.993	15.734	15.871	16.822					
FSMOGSA	12.002	12.399	13.118	14.723	14.892	15.177					
Ionosphere_Dataset											
GSA	20.400	20.113	17.391	14.764	16.430	16.412	17.023	17.321	17.110	18.021	
BPSO	24.032	20.132	17.943	12.730	12.023	11.899	11.071	14.132	16.234	16.942	
NSPSOFS	11.115	11.172	11.093	10.987	10.123	10.141	10.112	10.023	9.897	8.098	
FSMOGSA	12.976	11.132	10.398	9.341	9.076	8.991	9.112	9.354	9.310		
Vehicle_Dataset											
GSA	27.114	13.344	14.019	20.786	23.324	23.124	22.921				
BPSO	26.221	10.223	15.216	21.576	23.113	24.901	24.984				
NSPSOFS	22.223	16.912	17.445	18.139	18.939	18.175	17.945				
FSMOGSA	13.897	15.897	14.111	15.897	15.987	16.391	17.012				

Figure 2(a–d) show the graphical display of the experiment on each data set.

The FSMOGSA algorithm shows a great degree of stabilisation of the low error rate and minimising the number of features to achieve our objectives. Since our multi-objective was to maximise performance by reducing the error rate and at the same time reducing the number of features, we used the error function in Eq. (14) to achieve these goals. While the generation of new particles increased the chances of obtaining optimal sets of non-dominated solutions, the indexed sets facilitate the ranking and choosing of the best solution sets. The experimental validation of (FSMOGSA) is a

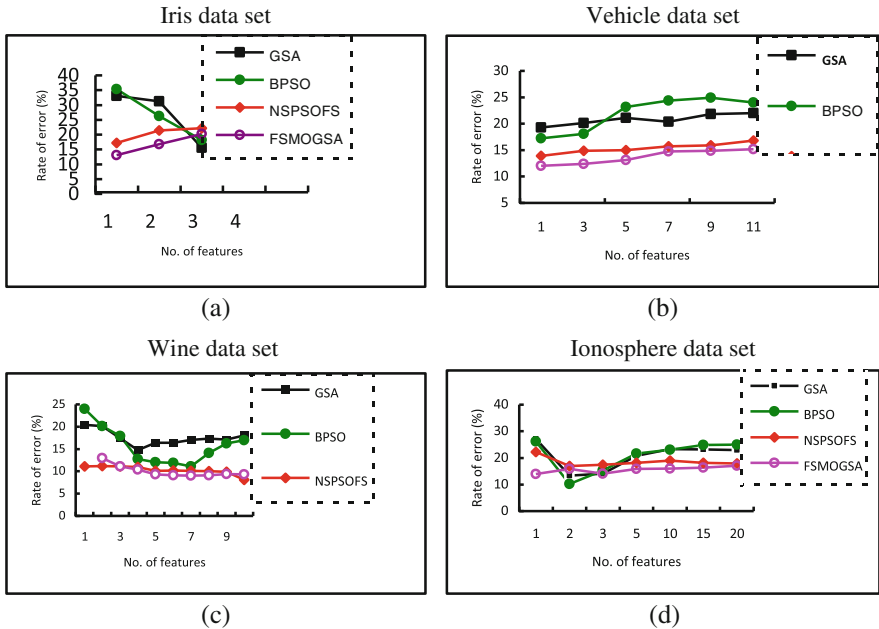


Fig. 2. (a) Is the Iris data set, the error rate was lower for FSMOGSA next by NSPSOFS, the features are fewer in this data set. For the data set; vehicle in (b), FSMOGSA still has the least error rate with reduced features. Then, the wine data set (c) the error rate is lower for NSPSOFS than our FSMOGSA but it is negligible and on the fourth data set (d), FSMOGSA showed the best result among the four algorithms employed on the Ionosphere data set.

good indication of the efficiency in its application to feature selection in a multi-objective task over most existing feature selection methods. Again, it indicates that FSMOGSA searches for the non-dominated solutions through the integration of minimal feature set numbers and classifier performance to yield optimised indexed non-dominated (Pareto fronts) solutions with high classification accuracy than the other three methods.

5 Conclusion

The experimental validation indicates that our method is a more efficient one. The best results for the multi-objective algorithm's validation were obtained by our FSMOGSA algorithm, the next best result was NSPSOFS both of which are hybrid methods. This shows that the hybrid methods have a better performance than the regular methods. From the experiments, FSMOGSA was noted to be quite unparalleled in comparison with the other methods in reducing the error rate and maximising the general performance by minimising the irrelevant features. For the objective of efficient performance we suggest a future work in binary GSA hybrid method.

Acknowledgements. This project is supported by the National Natural Science Foundation of China (61472161, 61133011, 61303132, 61202308), Science & Technology Development Project of Jilin Province (20140101201JC, 201201131).

References

1. Chen, B., Chen, L., Chen, Y.: Efficient ant colony optimization for image feature selection. *Sig. Process.* **93**, 1566–1576 (2013)
2. Jing, L., Zhang, C., Ng, M.K.: SNMFCA: supervised NMF-based image classification and annotation. In: *IEEE 2011* (2011)
3. Hamdani, T.M., Won, J.-M., Alimi, A.M., Karray, F.: Multi-objective feature selection with NSGA II. In: *Proceedings of 8th ICANNGA Part I*, vol. 4431, pp. 240–247 (2007)
4. Tian, H., Yuan, X., Ji, B., Chen, Z.: Multi-objective optimization of short-term hydrothermal scheduling using non-dominated sorting gravitational search algorithm with chaotic mutation. *Energy Convers. Manage.* **81**, 504–519 (2014)
5. Bhowmik, A.R., Chakraborty, A.K.: Solution of optimal power flow using non dominated sorting multi-objective opposition based gravitational search algorithm. *Electr. Power Energy Syst.* **64**, 1237–1250 (2015)
6. Xue, B., Zhang, M., Browne, W.N.: Particle swarm optimization for feature selection in classification: a multi-objective approach. *IEEE Trans. Cybern.* **43**(6), 1656–1671 (2013)
7. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: GSA: a gravitational search algorithm. *Inform. Sci.* **179**, 2232–2248 (2009)
8. Tian, H., Yuan, X., Ji, B., Chen, Z.: Multi-objective optimization of short-term hydrothermal scheduling using non-dominated sorting gravitational search algorithm with chaotic mutation. *Energy Convers. Manage.* **81**, 504–519 (2014)